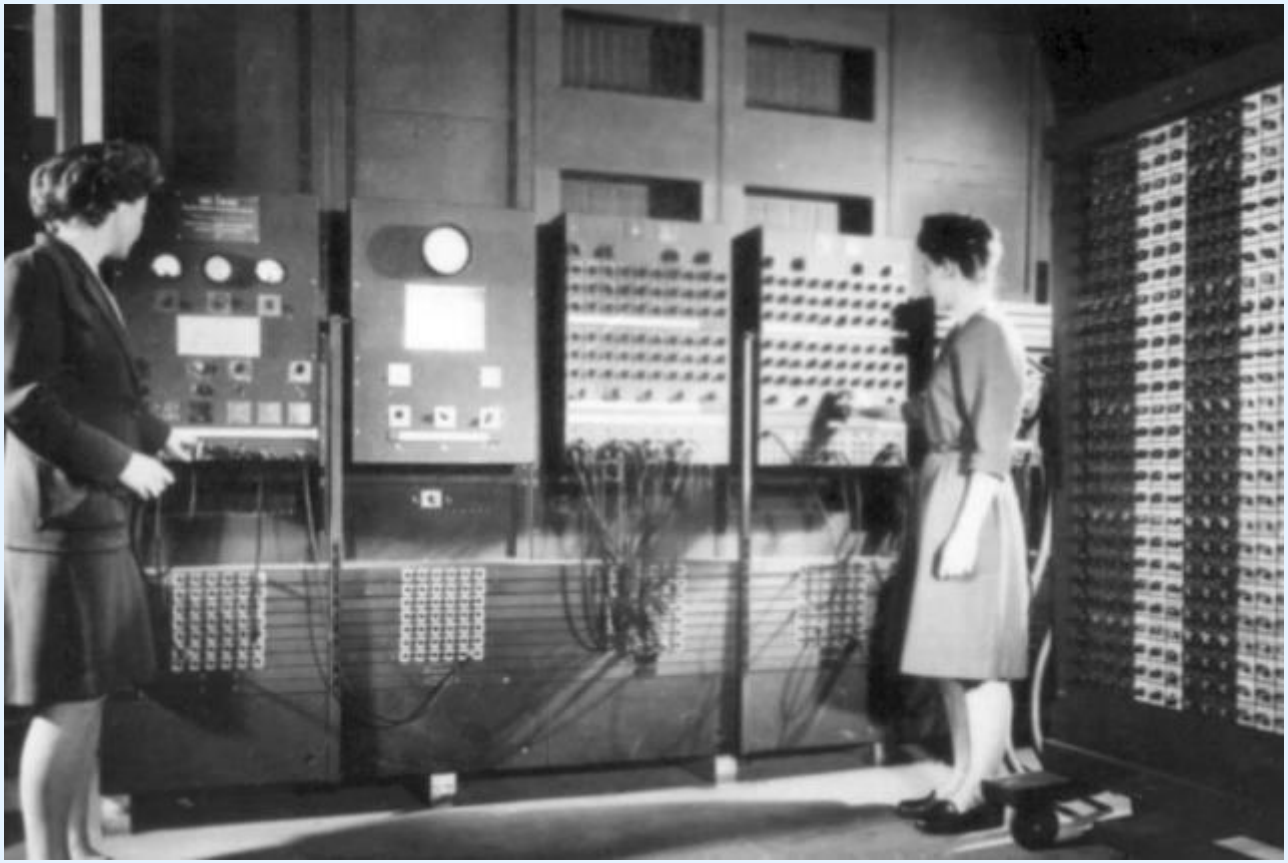
GCSE
Computer Science

Why study Computer Science

- You want to learn about how computers work
- **You want to learn how to program to computers**
- You like solving puzzles and thinking logically
- **You want to learn about how computer technology is evolving and changing**

Why is ENIAC from 1943 still a bit like…



ENIAC's programmers at work. On the left is Betty Jean Jennings, and Fran Bilas is on the right.

This…

GCSE CS – Edexcel Structure

Paper	How it is assessed?
Paper 1: Principles of Computer Science	Written exam 75 marks 1 hour 30 minutes (no calculators) Short questions (max. 6 marks)
Paper 2: Application of Computational Thinking	**On-screen exam** **75 marks** **2 hours** **6 Python coding questions**

Bletchley Park – The Enigma

Unit 1: Typical Exam Questions

1 Data

(a) Identify the smallest unit of measurement.

(1)

- [x] **A** bit
- [ ] **B** byte
- [ ] **C** kibibyte
- [ ] **D** nibble

Unit 1: Typical Exam Questions

(c) Routers send packets that contain data around the internet.

State **two other** items found in a packet.

(2)

The destination IP address

1 .....

.....

The sender IP address

2 .....

.....

Time Stamp, Packet Number, Error checking field

Unit 1: Typical Exam Questions

(e) A team of programmers is creating the code for an alarm system. The system uses a high-level programming language for the touchscreen graphical user interface and a low-level language for the control unit that monitors the sensors and triggers the alarm.

Discuss the characteristics of high-level languages and low-level languages that make them appropriate for the team of programmers to code these uses.

Your answer should consider:

- the purpose of the system
- the advantages of high-level languages
- the advantages of low-level languages.

(6)

Unit 1: Typical Exam Questions

Advantages of high-level languages:

- High-level languages come with libraries of ready-made graphical user interface components (buttons, icons and menus), which the team can use to reduce the amount of code they have to write from scratch.
- High-level languages have a range of integrated development tools, editors and syntax checkers, which will enable the team to develop the interface code more efficiently.

Advantages of low-level languages:

- Code written in assembly language normally executes more quickly and takes up less memory than code written in a high-level language. This may be crucial to enable the control unit for the alarm system to function effectively.
- There may be no high-level language for the microprocessor chip inside the control unit, so an assembly language would have to be used for it.

Unit 2: Typical Exam Question 1

A program simulates the roll of a dice. The program uses a random number generator to create a random integer, between 1 and 6, to represent the roll.

Open file **Q01**.

Amend the code to add or complete lines to:

- import the random library
- create one variable
- create one constant
- assign the result of a library call to a variable
- display a message and the contents of a variable on the screen.

Do **not** add any additional functionality.

Save your amended code file as **Q01FINISHED.py**

(Total for Question 1 = 7 marks)

Unit 2: Typical Exam Question 1

```
1  # ---------------------------------------------------------------------
2  # Import libraries
3  # ---------------------------------------------------------------------
4
5  # ===> Complete this line to import the random library
6  import
7
8  # ---------------------------------------------------------------------
9  # Global variables
10 # ---------------------------------------------------------------------
11
12 # ===> Create an integer variable named roll and set it to 0
13
14
15 # ===> Create a constant named SIDES and set it to 6
16
17
18 # ---------------------------------------------------------------------
19 # Main program
20 # ---------------------------------------------------------------------
21
22 # ===> Assign the result of this library call to the variable roll
23  = random.randint(1, SIDES)
24
25 # ===> Display the message "Your roll is" and the variable roll
26
```

Unit 2: Typical Exam Question 1

```
1  # ---------------------------------------------------------------
2  # Import libraries
3  # ---------------------------------------------------------------
4
5  # ===> Complete this line to import the random library
6  import random
7
8  # ---------------------------------------------------------------
9  # Global variables
10 # ---------------------------------------------------------------
11
12 # ===> Create an integer variable named roll and set it to 0
13 roll = 0
14
15 # ===> Create a constant named SIDES and set it to 6
16 SIDES = 6
17
18 # ---------------------------------------------------------------
19 # Main program
20 # ---------------------------------------------------------------
21
22 # ===> Assign the result of this library call to the variable roll
23 roll = random.randint(1, SIDES)
24
25 # ===> Display the message "Your roll is" and the variable roll
26 print ("Your roll is", roll)
27
```

Unit 2: Typical Exam Question 5

Students are collecting data about the amount of water needed to fill different sized paper cones. Their measurements are compared to a calculated volume.

The formula to calculate the volume of a cone is:

$$V = \frac{1}{3}\pi r^2 h$$

- V is volume
- π is the constant Pi
- r is the radius of the base of the cone
- h is the height of the cone.

Amend the program and subprogram to meet the following requirements:

- the subprogram must work for any values of radius and height passed as parameters. You can assume values passed to the subprogram will always be numbers. No validation is required
- the subprogram must calculate the volume based on the input parameters
- the main program must print the volume, formatted to show three decimal places (e.g. 16.135).

Do **not** add any additional functionality.

Save your amended code as **Q05FINISHED.py**

(Total for Question 5 = 15 marks)

Unit 2: Typical Exam Question 5

```
7  # -------------------------------------------------------------------
8  # Global variables
9  # -------------------------------------------------------------------
10
11 # Hard coded for testing
12 coneHeight = 10.7
13 baseRadius = 1.2
14 coneVolume = 0.0
15
16 # -------------------------------------------------------------------
17 # Subprograms
18 # -------------------------------------------------------------------
19 # ===> Add parameters inside the brackets
20 def calcVolume (                        ):
21
22     print ("The radius is:", pRadius)
23     print ("The height is:", pHeight)
24
25     # ===> Complete the calculation for the volume
26
27     print ("The volume is:", theVolume)
28
29     # ===> Return the volume to the caller
30
31
32 # -------------------------------------------------------------------
33 # Main program
34 # -------------------------------------------------------------------
35
36 # ===> Call the subprogram, passing parameters,
37 #      and catch the returned value in the correct variable
38
39
40 # ===> Print the total volume to three decimal places using string.format()
41 # ===> by completing the pattern inside the { }
42 print ("{                }".format(coneVolume))
```

Unit 2: Typical Exam Question 5

```
16 # ---------------------------------------------------------------------
17 # Subprograms
18 # ---------------------------------------------------------------------
19 # ===> Add parameters inside the brackets
20 def calcVolume (pRadius, pHeight):
21
22     print ("The radius is:", pRadius)
23     print ("The height is:", pHeight)
24
25     # ===> Complete the calculation for the volume
26     theVolume = 1/3 * math.pi * math.pow (pRadius, 2) * pHeight
27     # theVolume = 1/3 * math.pi * pRadius**2 * pHeight
28     # theVolume = 1/3 * math.pi * pRadius * pRadius * pHeight
29
30     print ("The volume is:", theVolume)
31
32     # ===> Return the volume to the caller
33     return (theVolume)
34
35 # ---------------------------------------------------------------------
36 # Main program
37 # ---------------------------------------------------------------------
38
39 # ===> Call the subprogram, passing parameters,
40 #      and catch the returned value in the correct variable
41 coneVolume = calcVolume (baseRadius, coneHeight)
42
43 # ===> Print the total volume to three decimal places using string.format()
44 # ===> by completing the pattern inside the { }
45 print ("{:.3f}".format(coneVolume))
```

What will you study?

- Python programming and skills in algorithm building
- **Computing science theory topics such as networking, cybersecurity, how computers are made and what their different components do**

What are the lessons like

- Unit 1 are classroom-based theory lessons – you will be doing lots of exercises and challenges to build up your knowledge and skills on topics such as Networking, Computational Thinking and Data
- **Unit 2 is a computer-based lesson (with some theory). It is mostly hands on learning how to program in Python and solve programming problems.**

What is the only way to survive this game?

```
print ("You are in a dark cave. There are three ways you can go.")

direction = input ("Which way would you like to go? (Left, Right or Forward) ")

if direction == "L":
    print ("You are eaten by a mega spider.")
elif direction == "R":
    print ("A baby dragon turns you into toast.")
elif direction == "F":
    print ("You crawl out of the cave through a low tunnel.")
else:
    print ("You can't work out what to do. A bear comes and sits on you.")
```

What happens if you enter anything apart from L, R or F?

This is a different version – what's different? What will happen?

```
print ("You are in a dark cave. There are three ways you can go.")

direction = input ("Which way would you like to go? (Left, Right or Forward) ")

possibleDirections = ["L", "R", "F"]

while direction not in possibleDirections:
    print ("You can't go that way!")
    direction = input ("Which way would you like to go? (Left, Right or Forward) ")

if direction == "L":
    print ("You are eaten by a mega spider.")
elif direction == "R":
    print ("A baby dragon turns you into toast.")
elif direction == "F":
    print ("You crawl out of the cave through a low tunnel.")
```

A typical online Unit 2 activity

Predict

```
1 # global variables
2 scores = [23, 19, 10, 30]
3
4 # main program
5 print (scores)
6 print (len (scores))
7 print (scores[2])
8 print (scores[3])
```

Template | Model Solution | Options

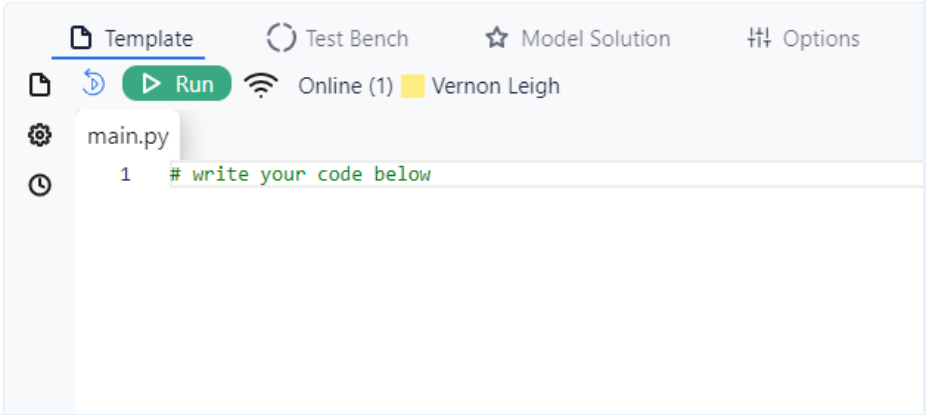
Write what you think this program will do and output

B i S U </> H1 H2 H3 ☰ ☰ ☰ ❝

Default response

Run and Investigate

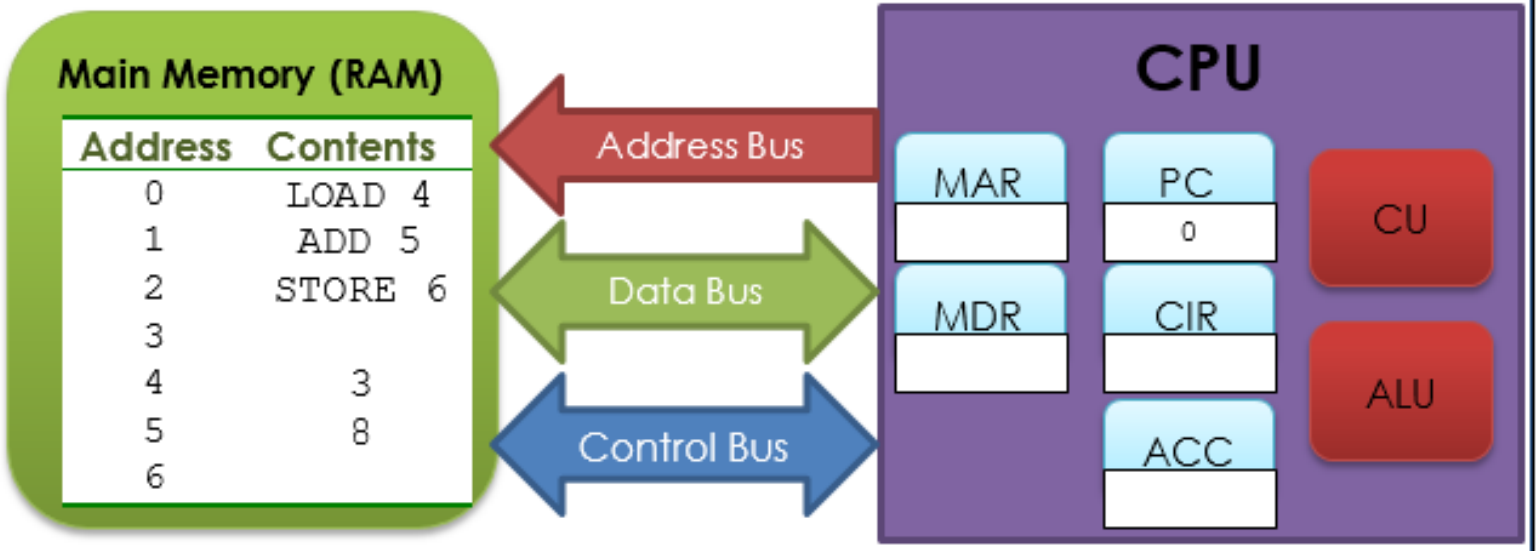
1. Copy / paste or write the code above into the IDE below and try it out.
2. Change the code so it gives more user-friendly messages (e.g. "The list size is ...")
3. Add code that:
   a. Asks the user for the index position of the score they want
   b. Displays that score
4. Add a second list that stores 4 player names (of your choice) and then prints them out.



Wallingford SCHOOL 1659

A typical classroom Unit 1 task

The contents of main memory and the CPU registers are currently:



a) What is the data in memory address 4?

b) What is the instruction in memory address 2?

c) What is the purpose of the instructions above?

Resources we use

- Craig N Dave – teaching videos specific to Edexcel Computer Science
- **Pearson Revision Guides and Workbooks**
- Thonny and IDLE – free Python programming environments
- **CodingRooms – an online site for teaching and learning coding skills**

How you find out more?

- Speak to the subject lead: Mr Leigh
- **Email Mr Leigh at leighv@wallingfordschool.com**
- Speak to older students who are already taking the course

Results (Summer 2022)

- 86% 4+
- 65% 5+
- 24% 7+